Now you came all the way to use Microsoft Computer Vision API. We will talk about how to use Analyze image API first. With Analyze image API, we can extract visual features based on image content provided by the user. It gets

Below is an official documentation(reference) for Analyze image API.

<https://westus.dev.cognitive.microsoft.com/docs/services/56f91f2d778daf23d8ec6739/operations/56f91f2e778daf14a499e1fa>

Let’s see how it works first seeing how Request URL looks like.

When you request to server, Request URL goes like this

[https://[location].api.cognitive.microsoft.com/vision/v1.0/analyze[?visualFeatures][&details][&language](https://[location].api.cognitive.microsoft.com/vision/v1.0/analyze%5b?visualFeatures%5d%5b&details%5d%5b&language)**]**

**you choose location among westus, eastus2, westcentralus, westeurope, and southeastasia.**

**As request parameters there are visualFeatures, details, and language. VisualFeatures contain Categories, Tags, Description, Faces, ImageType, Color, and Adult.**

**details contain Celebrities, and Landmarks**

**language has two options like en; english, zn; simplified chinese.**

I tested some images with this console. And my request was like this

https://southeastasia.api.cognitive.microsoft.com/vision/v1.0/analyze?visualFeatures=Tags, ImageType, Color, Adult, Faces&details=Celebrities, Landmarks&language=en

You can see how request URL with various parameters should be like referring above request URL ,which really works in console.

I posted a URL of Bill Gates standing next to Eiffel tower (<http://s1.lprs1.fr/images/2011/04/04/1393033_gates-trocadero_1000x625.jpg> ) to test the console.



I got a result like this from the picture above.

This is the response in JSON form with the former request.

{ "categories": [ { "name": "people\_", "score": 0.97265625, "detail": { "celebrities": [ { "name": "Bill Gates", "faceRectangle": { "left": 427, "top": 184, "width": 104, "height": 104 }, "confidence": 0.9998789 } ] } } ], "adult": { "isAdultContent": false, "isRacyContent": false, "adultScore": 0.0095563493669033051, "racyScore": 0.013779528439044952 }, "tags": [ { "name": "person", "confidence": 0.99934250116348267 }, { "name": "sky", "confidence": 0.9953383207321167 }, { "name": "outdoor", "confidence": 0.99214887619018555 }, { "name": "man", "confidence": 0.91900533437728882 }, { "name": "suit", "confidence": 0.76639425754547119 }, { "name": "people", "confidence": 0.66376322507858276 }, { "name": "dressed", "confidence": 0.336028516292572 }, { "name": "older", "confidence": 0.2262205183506012 } ], "requestId": "57e878ef-127d-4aa8-b5ac-fbd3117943d4", "metadata": { "width": 1000, "height": 625, "format": "Jpeg" }, "faces": [ { "age": 63, "gender": "Male", "faceRectangle": { "left": 427, "top": 184, "width": 104, "height": 104 } } ], "color": { "dominantColorForeground": "White", "dominantColorBackground": "White", "dominantColors": [ "White", "Black", "Brown" ], "accentColor": "85673C", "isBWImg": false }, "imageType": { "clipArtType": 0, "lineDrawingType": 0 } }

You can find some parameters I put on the request like Celebrites, adult, etc. Other features like categories and metadata comes out as a result default. The Server could not detect the Eiffel tower because it quite looks blur but detected Bill gates exactly. The response is in the form of JSON. So you can extract information you want to extract easily as you already know how to deal with JSON.

{ "categories": [ { "name": "people\_", "score": 0.97265625, "detail": { "celebrities": [ { "name": "Bill Gates", "faceRectangle": { "left": 427, "top": 184, "width": 104, "height": 104 }, "confidence": 0.9998789 } ] } } ], "adult": { "isAdultContent": false, "isRacyContent": false, "adultScore": 0.0095563493669033051, "racyScore": 0.013779528439044952 }, "tags": [ { "name": "person", "confidence": 0.99934250116348267 }, { "name": "sky", "confidence": 0.9953383207321167 }, { "name": "outdoor", "confidence": 0.99214887619018555 }, { "name": "man", "confidence": 0.91900533437728882 }, { "name": "suit", "confidence": 0.76639425754547119 }, { "name": "people", "confidence": 0.66376322507858276 }, { "name": "dressed", "confidence": 0.336028516292572 }, { "name": "older", "confidence": 0.2262205183506012 } ], "requestId": "57e878ef-127d-4aa8-b5ac-fbd3117943d4", "metadata": { "width": 1000, "height": 625, "format": "Jpeg" }, "faces": [ { "age": 63, "gender": "Male", "faceRectangle": { "left": 427, "top": 184, "width": 104, "height": 104 } } ], "color": { "dominantColorForeground": "White", "dominantColorBackground": "White", "dominantColors": [ "White", "Black", "Brown" ], "accentColor": "85673C", "isBWImg": false }, "imageType": { "clipArtType": 0, "lineDrawingType": 0 } }

Ok, now we are done with console. Let’s jump into API code written in JavaScript to make a useful app for us.

JavaScript part of the program starts with lines below.

$(function() {

$(form).submit(function(event){}

I made a form to get a URL as an input. I need to DOM elements of the page are all loaded and ready to use that. So I put $(function(){}

$(function() is a short version of

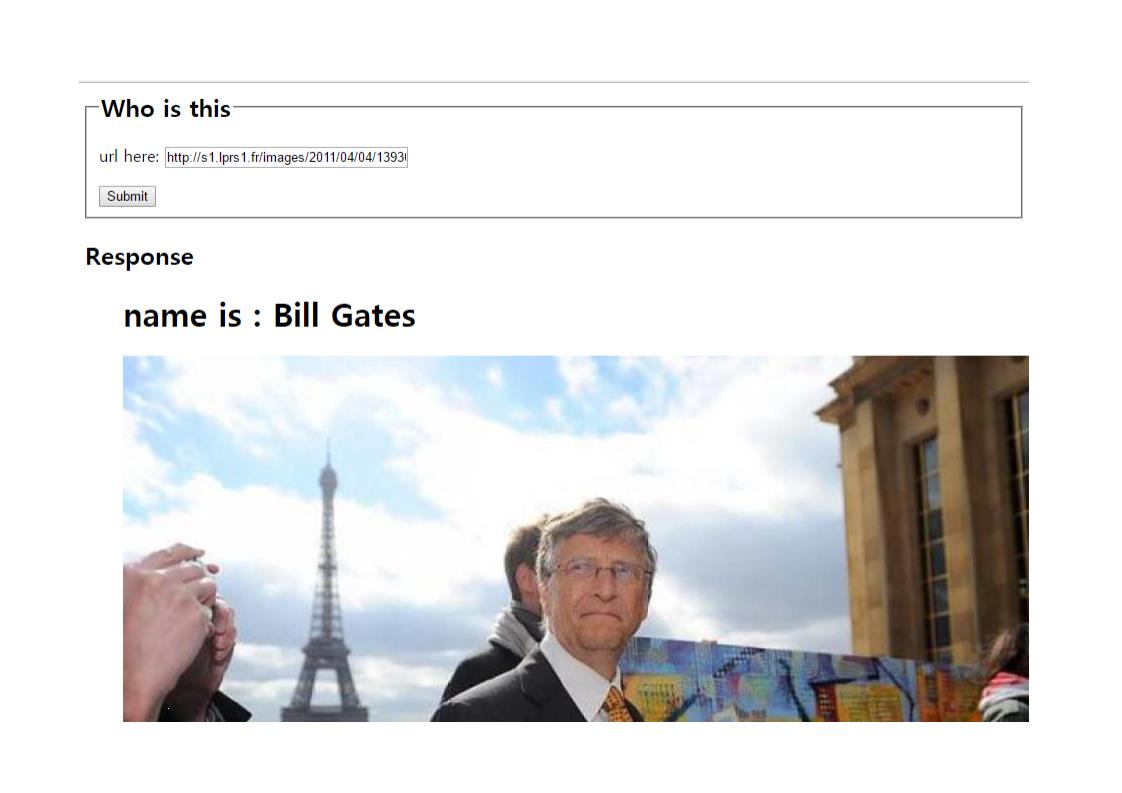
$(document).ready(function() { ... });

and after I put $(form).submit(function(event){}. This means like “Do something inside the bracket when a form is submitted.

I put my API key, and input URL, and payload(myBody here) to variables.

Parameters are stored in an object called params.

I put $.ajax({}) to do an asynchronous AJAX request. Remember you had to write long codes before when you used pure JavaScript. You can apply this form of AJAX to other APIs later when you use jQuery. Inside $.ajax({}), you can see processes to request data to url and before sending we set request header with .setRequestHeader() function. And when the request is successfully doen, .done() part works and the response is stored in data parameter while there is something wrong .fail() part works so it put “error” alert on the browser. You can deal with information deduced from the picture from response which is stored in a form of JSON.



Manipulating JSON I could make a simple application showing the name of the celebrities.

Try it now own your own. Try only picture with the celebrity is in. If not, the program will not work properly. It will only show you the picture from the URL you input.

Now I will end the explanation about this analyze image API.